Privacy Enhancing Technologies
Lecture 8 – Data Privacy Beyond Cryptography

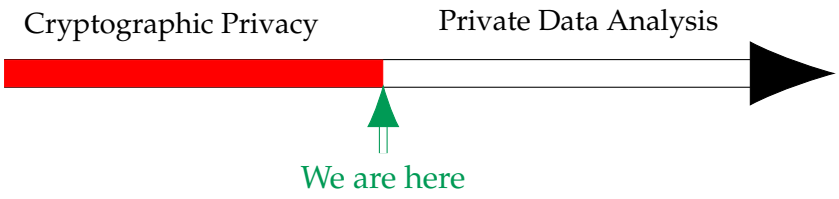
Florian Tramèr

These lecture notes are inspired by those of Gautam Kamath.[1]

AGENDA

1. Recap on cryptographic privacy
2. When is cryptography not enough?
3. Data reconstruction attacks
4. Reconstruction attacks in practice

1 Where Are We?



We've reached the midpoint of this class. So far, we have focused on cryptographic approaches to privacy. Informally, these approaches all aim to compute some function f over multiple parties' data, while revealing *no more information than the output of* f.

But what if this isn't enough? What if the output of the function itself reveals sensitive information about the data of one or more parties? In particular, what happens if we start computing *many* functions f over parties' data?

Here we enter the realm of *private data analysis*, which will be our focus for the rest of this course. From now on, we will primarily focus on *statistical* (a.k.a. information-theoretic) notions of privacy. We will still consider adversaries, but these adversaries will not be computationally bounded.

2 When Aggregation Isn't Private

Suppose we collect aggregate statistics, as in the previous lecture. But we're going to do this for many different statistics of interest.

This situation occurs frequently in practice: e.g., a survey might ask many different questions from the same group of people and report the aggregate results for each question. Scientific studies (e.g., in genomics) often report aggregate statistics for many different traits

[1] http://www.gautamkamath.com/CS860notes/lec2.pdf

or genetic markers. Below, we'll consider an illustrative example of a *census*, which collects demographic information of a population.

A census (e.g., as conducted in the US every ten years) might collect individual information such as age, gender, ethnicity, marital status, etc. As this data may be too sensitive to release directly, the census aggregates this information across different subsets of a population, and publicly releases only those aggregates.

Small counts and k-anonymity. A first privacy issue that can arise with aggregate data is that a statistic may be computed over a small number of people. For example, suppose that there is only one person in the data who is a Black adult over 64. Then, releasing "aggregate" statistics for this category immediately reveals other information about this one individual (e.g., their marital status).

A popular attempt at preventing this is to merge or suppress statistics that are computed over only a few people. Probably the most well-known approach is *k-anonymity* [Swe02], which (informally) ensures that each individual's data is indistinguishable from at least k-1 other individuals' data. We won't cover k-anonymity in detail in this course, as it suffers from similar issues as the approach we discuss in the following.

As a (highly) simplified example, consider the census information below collected from a small "block" of 7 people. We aggregate their average and median age, further broken down by gender, ethnicity, and marital status. Any statistic that is computed over only 1 or 2 people is suppressed (marked as (D)).

			Age	
Statistic	**Group**	**Count**	**Median**	**Mean**
1A	Total Population	7	30	38.0
2A	Female	4	30	33.5
2B	Male	3	30	44.0
2C	Black or African American	4	51	48.5
2D	White	3	24	24.0
3A	Single Adults	(D)	(D)	(D)
3B	Married Adults	4	51	54.0
4A	Black or African American Female	3	36	36.7
4B	Black or African American Male	(D)	(D)	(D)
4C	White Male	(D)	(D)	(D)
4D	White Female	(D)	(D)	(D)
5A	Persons Under 5 Years	(D)	(D)	(D)
5B	Persons Under 18 Years	(D)	(D)	(D)
5C	Persons 64 Years or Over	(D)	(D)	(D)

Table 1: Fictitious statistics of a Census block, from [GAM19]. (D) indicates that the statistic has been suppressed because it is an aggregate from only 1 or 2 people.

While suppressing statistics hides some information, it still allows inferring details about individuals. For example, since we know that there are 4 Black people, and 3 of them are female, we can infer that there is one Black male. As the average age for this person was suppressed, we cannot learn it directly. But here we just looked at logical combinations of two statistics. What if we go beyond that?

Constraint satisfaction. The different statistics computed over the same population give us a lot of redundant information. It turns out that we can use this redundancy to recover

all information about the population!

Let's start with an example:

> From statistic 2B, we know there are 3 males. Denote their ages as $A \leq B \leq C$.
>
> Assuming ages are integers between 0 and 125, this would mean there could be as many as $126^3 \approx 2$ million possible combinations of A, B, C.
>
> But we know that their average age is 44 and the median age is 30. This immediately tells us that $B = 30$, $A \leq 30$, and $C \geq 30$.
>
> We then have $(A + B + C)/3 = 44$, or equivalently $A + C = 102$. This reduces the number of combinations to just 31.

So by looking at a single constraint, we recovered the exact age of one person and reduced the number of possible combinations for the other two by 99.99%.

We can automate this process by writing out constraints for all statistics computed over the same population, and then solving the constraint system to find a solution. Depending on the type of constraints we use, this problem is NP-Hard. But real-world data is rarely worst-case and there are a number of heuristic solvers that can tackle such problems with millions of constraints.

For the statistics in the census example above, it turns out that a single solution exists, which can be found on a laptop in 0.1 seconds [GAM19].

3 How Many Queries Can You Answer Privately?

So what went wrong with the above data release? Essentially, it's a matter of releasing *too many*, *highly accurate* statistics about the population.

We will now work towards a formal model of data analysis that will allow us to prove bounds on what can and cannot be computed privately. This model will be rather simplistic, but it will allow us to prove some interesting impossibility results.

The curator model. We will consider a setting where a trusted party (the "curator") collects a database of records. The adversary gets to ask a series of *queries* over this database, which the curator answers (possibly with noise).

The database consists of n records (or "rows"), each of which has multiple attributes (the "columns"). We'll assume that all but one of these attributes are public (i.e., they are known to the adversary). The last attribute is a single private bit. For example:

	Public attributes		*Private attribute*
Name	**Age**	**Gender**	**Disease?**
Alice	25	F	1
Bob	30	M	0
…	…	…	…
Zack	18	M	1

The adversary will be allowed to ask *subset queries* over this database, which sum up the private attribute for a subset of the rows. For example, the adversary might ask:

> *"How many women under 25 in the database have the disease?"*

Or in (pseudo)-SQL form:

```
SELECT SUM(Disease) FROM database WHERE Gender = F AND Age < 25;
```

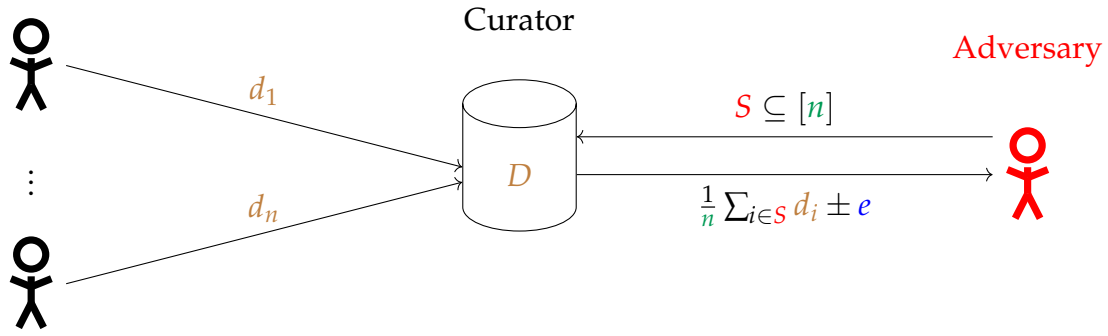
We will abstract this away as follows:

- We identify the database with the private attribute, i.e., $D = \{d_1, d_2, \ldots, d_n\} \in \{0,1\}^n$.
- A query from the adversary is a subset of rows, $S \subseteq [n]$.
- The true answer to the query is the *normalized*[2] sum of the private attribute for the rows in S, i.e., $\texttt{Ans}(S) = \frac{1}{n}\sum_{i \in S} d_i$.
- The curator answers the query by returning $\texttt{Noisy}(S)$. This may not be the exact true answer, but a "noisy" version.[3] We denote the amount of noise as

$$e = |\texttt{Noisy}(S) - \texttt{Ans}(S)|.$$

  We assume that the noise is bounded, i.e., $e \leq E$, but is otherwise arbitrary (in particular, the noise is not necessarily independent across queries. It might not even be random. All we know is that the noise is bounded).

The picture below illustrates this setup.



So let's see what's possible in this setting. That is, how many queries can we answer, and with what amount of noise, before we start leaking "too much" information to the adversary?

How do we define what it means for the adversary to learn "too much" information? For now, we'll content ourselves with a very weak goal: we want to make sure that the adversary cannot *perfectly reconstruct a large chunk of the database*.

[2] This normalization just scales all results by $1/n$ and is not really important for the analysis. But it will make things easier when we look at some results from differential privacy.

[3] If there were no noise, the adversary could easily reconstruct the i-th bit of D by asking the query $S = \{i\}$.

Definition 1 (Blatant *non*-privacy. [DN03])**.** The curator's algorithm `Noisy` is *blatantly non-private* if the adversary can, with *high probability* $1 - o(1)$,[a] construct a database $\texttt{guess} \in \{0,1\}^n$ such that D and $\texttt{guess}$ differ in $o(n)$ coordinates.

[a] That is, the probability goes to 1 as the database size n goes to infinity.

A blatantly non-private algorithm is pretty disastrous for privacy (hence the name). It means that the adversary can reconstruct more than any constant fraction (e.g., 99.99%) of the database exactly. We definitely don't want that...

A celebrated set of results of Dinur and Nissim [DN03] shows that if we want to answer many queries and avoid blatant non-privacy, then we necessarily need to add a large amount of noise. These results then paved the way to the formalization of differential privacy, which we will discuss in the next few lectures, and which provides a more granular notion of privacy leakage.

2^n subset queries cannot be answered privately with $o(1)$ noise. The first result of Dinur and Nissim considers an adversary who can ask a very large number of queries, *exponential* in the size of the database (actually every single subset query!).

Theorem 1 (Dinur and Nissim, 2003)**.** If the adversary can ask 2^n subset queries on a database of size n, and the curator adds noise of size at most E to each query, then the adversary can output a database $\texttt{guess} \in \{0,1\}^n$ such that D and $\texttt{guess}$ differ in at most $4En$ positions.

As an example, suppose we set $E = 1/401$. This is a rather large amount of noise: for any query on a subset of less than $1/400$-th of the data, the noise will drown out any signal. And yet, even with this large amount of noise, an adversary can reconstruct the database on 99% of the entries. And if we set a smaller noise, $E = o(1)$, then the curator's algorithm is blatantly non-private!

Proof. The adversary's algorithm is very simple: make queries for all 2^n subsets $S \subseteq [n]$. Then, loop over all possible databases $\texttt{guess} \in \{0,1\}^n$ and output the first one that is *consistent* with all the queries, i.e.,

$$\left| \frac{1}{n} \sum_{i \in S} \texttt{guess}_i - \texttt{Noisy}(S) \right| \leq E \quad \forall S \subseteq [n].$$

First, it's easy to see that this algorithm always terminates, since the true database D is consistent with all queries.

Now, let's show that the guess output by the adversary differs from the true database D in at most $4En$ positions, i.e.,

$$\texttt{dist}(D, \texttt{guess}) := \sum_{i=1}^{n} \mathbb{1}[d_i \neq \texttt{guess}_i] \leq 4En.$$

We proceed by contradiction. Suppose that $\texttt{dist}(D, \texttt{guess}) > 4En$. We now split the errors into two parts:

$$q_0 = \{i \mid d_i = 0, \texttt{guess}_i = 1\} \quad \text{and} \quad q_1 = \{i \mid d_i = 1, \texttt{guess}_i = 0\}.$$

Since we have that $|q_0| + |q_1| = \texttt{dist}(D, \texttt{guess}) > 4En$, one of the two sets must be larger than $2En$. Without loss of generality, let's assume that $|q_0| > 2En$.

We have that $\frac{1}{n}\sum_{i \in q_0} d_i = 0$. And so it must be that $\texttt{Noisy}(q_0) \leq E$. On the other hand, $\frac{1}{n}\sum_{i \in q_0} \texttt{guess}_i = \frac{|q_0|}{n} > 2E$. And so we have

$$\left| \frac{1}{n} \sum_{i \in q_0} \texttt{guess}_i - \texttt{Noisy}(q_0) \right| > E \,,$$

which contradicts the fact that guess is consistent with all queries. □

n subset queries cannot be answered privately with $o(1/\sqrt{n})$ noise. To recap, the previous result shows that if the adversary can ask all 2^n subset queries, then we need noise of magnitude $\Omega(1)$ to prevent the adversary from reconstructing essentially the entire database. This is pretty bad: the only way to prevent a flagrant breach of privacy is to add noise that essentially drowns out any signal (the true answer is always in the range $[0, 1]$). But this requires the adversary to make an exponentially large number of queries, which would be impractical except for very small databases.

The next result shows that if the adversary only asks a *linear* number of subset queries, then noise of magnitude $\Omega(1/\sqrt{n})$ is necessary for privacy.

> **Theorem 2** (Dinur and Nissim, 2003)**.** If the adversary can ask n subset queries on a database of size n, and the curator adds noise of size bounded by $E = o(1/\sqrt{n})$ to each query, then the curator's algorithm is blatantly non-private.

Proof. The adversary's algorithm is similar as before, except that the adversary now makes n *random* subset queries (i.e., on vectors S with each entry being independent coin flips). The adversary then outputs a database guess that is consistent with all queries. This can be done by solving a *linear program* over the unknown variables d_i (the private bits of the database). The attack thus runs in polynomial time.

We won't formally prove that the adversary's output is close to D, but we'll give some intuition. We'll switch here to a rescaled setting where the database D, query S and guess guess are vectors in $\{-1, 1\}^n$ (where $S_i = 1$ if $i \in S$). The true answer is $\texttt{Ans}(S) = \frac{1}{n} S \cdot D$, and the curator returns a noisy version $\texttt{Noisy}(S)$ of this answer.[4] The idea is that if guess is far from the true database D, then for a random query S, the answers $\frac{1}{n} S \cdot D$ and $\frac{1}{n} S \cdot \texttt{guess}$ will be sufficiently far that they cannot be reconciled with noise of magnitude $o(1/\sqrt{n})$.

Consider the value $\Delta = (\texttt{guess} - D) \cdot S$, where S is a uniformly random query. Suppose that guess and D differ in $\Omega(n)$ positions. Then the value Δ is the sum of $\Omega(n)$ independent values ± 2, which is a rescaled and shifted version of a $\texttt{Binomial}(\Omega(n), 1/2)$. Notably, Δ has mean 0 and standard deviation $\Omega(\sqrt{n})$, and by *anti-concentration* of the Binomial distribution, it will take a value of magnitude $\Omega(\sqrt{n})$ with high probability. Thus the difference between $\frac{1}{n} S \cdot D$ and $\frac{1}{n} S \cdot \texttt{guess}$ is at least $\Omega(1/\sqrt{n})$.

So guess cannot be consistent with $\texttt{Noisy}(S)$ if the amount of noise is $o(1/\sqrt{n})$. With a bit more work (see [DN03]), we can show that *every* guess that is far from D will be ruled out with high probability, and so the guess output by the adversary must be close to D. □

[4] Note that here the query no longer corresponds to taking a sum over a subset of entries, but more generally to computing an arbitrary dot product between D and S.

4 Linear Reconstruction Attacks in Practice

These attacks are quite devastating, but the setup might seem somewhat unrealistic. The sensitive information of each user is a single bit, and the attacker can make arbitrary subset queries over the database (this somehow implies the attacker knows all the public contents of the database).

It turns out these are just artifacts of the relatively simple formal model we were using, which enables relatively clean proofs. Variants of these attacks do work in practice, and have been used to reconstruct real databases. Let's look at some examples:

The Aircloak challenge. In 2017, Aircloak—a German data anonymization company—ran a bug bounty challenge to reconstruct parts of a database by interacting with their "privacy preserving" query system *Diffix*. This system used a bunch of heuristics to add noise to queries, with the amount of noise increasing with how "specific" the query was (i.e., how many different attributes were used in the query).

In a nice attack, Cohen and Nissim [CN18] showed that an analyst could craft queries that only used a small number of attributes (and thus resulted in low noise), and that would sum up the values of a (roughly) random subset of the database. This then allowed them to apply the attack from Theorem 2 to reconstruct nearly the entire database (and win $5'000)!

The US Census. In Section 2 we looked at a fictitious example of a reconstruction attack for census data. However, researchers at the US Census have actually simulated such attacks on full census data, and found that their attacks could reconstruct the individual private responses of a large fraction of the US population (17–58% depending on the attacker's prior knowledge) [Abo21].

These results have proved to be controversial, however. Since this reconstruction is not "blatantly non-private", some researchers have argued that one could reconstruct a similarly large, *constant* fraction of the database by just picking attributes at random according to the population distribution (although under some rather *dubious* assumptions...) [Rug21]. Further research has since shown that reconstruction attacks significantly outperform random guessing, especially in small census blocks [DDK+23]. The possibility of such attacks have motivated the US Census bureau to adopt differential privacy for their data releases.

References

[Abo21] John M Abowd. Declaration of John M Abowd in case no. 3:21-cv-00211-rah-ecm-kcn, the State of Alabama v. United States Department of Commerce (2021). https://www.documentcloud.org/documents/21018464-fair-lines-america-foundation-july-26-2021-declaration-of-john-m-abowd, 2021.

[CN18] Aloni Cohen and Kobbi Nissim. Linear program reconstruction in practice. *arXiv preprint arXiv:1810.05692*, 2018.

[DDK+23] Travis Dick, Cynthia Dwork, Michael Kearns, Terrance Liu, Aaron Roth, Giuseppe Vietri, and Zhiwei Steven Wu. Confidence-ranked reconstruction of census microdata from published statistics. *Proceedings of the National Academy of Sciences*, 120(8):e2218605120, 2023.

[DN03] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM Symposium on Principles of Database Systems*, pages 202–210, 2003.

[GAM19] Simson Garfinkel, John M Abowd, and Christian Martindale. Understanding database reconstruction attacks on public data. *Communications of the ACM*, 62(3):46–53, 2019.

[Rug21] Steven Ruggles. Professor Steven Ruggles expert report in case no. 3:21-cv-00211-rah-ecm-kcn, the State of Alabama v. United States Department of Commerce (2021). https://users.pop.umn.edu/~ruggles/censim/Ruggles_Report.pdf, 2021.

[Swe02] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International journal of uncertainty, fuzziness and knowledge-based systems*, 10(05):557–570, 2002.